

Migrate a module

1. Create BT task in Internal Odoo.
2. Create branch 16.0.<task_code> from 16.0.
3. Execute **migration_script** with the module desired and the branch just created.


```
python3 ~/script_move_modules.py --modules turbo_swissdec --source_repo git@github.com:<owner>/<repo>.git --source_branch 10.0 --target_repo git@github.com:<owner>/<repo>.git --target_branch 16.0.<task_code>
```
4. Use the OCA **odoo-module-migrate** tool.


```
odoo-module-migrate --directory <path> --modules <module_1,module_2,module_3> --init-version-name 10.0 --target-version-name 12.0 --no-commit
```
5. Migrate following OCA guidelines. Example: Migration to V13.
6. Install in local and fix all problems.
7. Add docstrings and documentation.
8. Define unit tests.
9. Update README.
10. Commit and Push, create PR and assign to review it.

README

Command `python3 -m maintainer-tools.tools.gen_addon_readme --org-name "braintec" --addon-dir /home/moca1/BraintecProjects/<project_folder>/repos/<project>/<module> --repo-name "<project>" --no-commit --branch <main(16.0)>`

DESCRIPTION (MANDATORY) It explains what the module is about.

ORIGINAL_AUTHORS (MANDATORY) Who has contributed to an existing version.

CONTRIBUTORS Who has initially being involved in the first design

CONFIGURE How to configure the module before using it.

Do **NOT** forget to add the **BT icon**.

Git Submodules

Add `git submodule add -b <branch> git@github.com:<owner>/<repo>.git ext/<repo>`

Update `git submodule update <--options>`

Update options:

--init: Initializes submodules, cloning them and setting them to a specific commit if they haven't been initialized before.

--force: Forces submodule update, ignoring the commit information stored in the main repository and ensuring that submodules are updated.

--remote: Updates submodules to the latest commit of their remote branch, useful when submodules track a remote branch for updates.

Docker Deploy

Build `bt deploy build -s <server> -p <project>:<branch>`

Update `bt deploy update -s <server> -p <project>:<branch> -d <DB>`

Code checks

`bt test code-quality <module>`

Docker Shell

`bt docker shell`

Solve merging conflict

`resolve_conflicts_step_1 <origin_branch> <dest_branch>`

`resolve_conflicts_step_2 <origin_branch> <dest_branch>`

`resolve_conflicts_step_3 <dest_branch>`

Add script from Fixing conflicts in Test PR into **\$HOME/.bashrc**.

Then execute `. $HOME/.bashrc` in terminal to apply changes.

PyCharm Run configuration

Run `odoo-bin -d <DB> -i <install_modules> -u <update_modules> --i18n-overwrite`

Test `odoo_bin -d <DB> -i <modules> --test-tags=<modules> --stop-after-init`

Download Server DB

Command `bt docker load-data <server>`

-live Retrieves the current status of the DB on server.

-skip-tables Skips unnecessary tables like messaging and also filestore.

-clean-script Allows to indicate a certain clean script as a parameter.

BT Translations

1. Install `bt_translation` Add `bt_utils` module to your project and install `bt_translation`.
2. Export translations Check **Use BT Export**.
on Settings app
3. Compare both PO files On PyCharm, click on `bt_exported.po` and `.po` files and press `Ctrl+D`.
4. Fix them and repeat the process until they are identical.



By **carlosma7**
cheatography.com/carlosma7/

Not published yet.
Last updated 12th December, 2023.
Page 2 of 2.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
<https://apollopad.com>